

## Software Testing Homework 2

312551098 林佑庭

(a). List all of the input variables, including the (abstract) state variables.

Input variables:

- capacity
- element

State variables:

- BoundedQueue.capacity
- BoundedQueue.elements
- BoundedQueue.size
- BoundedQueue.front
- BoundedQueue.back

(b). Define the characteristics of the input variables. Make sure you cover all input variables.

Methods	Params	Returns	Values	Exception	Ch ID	Characteristic	Covered by
BoundedQueue	int	-	>=0		C1	Maximum number of elements of the queue	
				Invalid value			C1
enqueue	number	-	-	Invalid type	C2	Check if the input type is a number or not	
				Invalid value	C3	Check if the input type is NaN or not	
				Illegal state			C6
dequeue	- (state)	number	number		C4	Pop the front element	
				Illegal state			C5
is_empty	- (state)	boolean	T, F		C5	The size of the queue is equal	

						to 0	
is_full	- (state)	boolean	T, F		C6	The size of the queue is equal to the capacity	

(c). Partition the characteristics into blocks. Designate one block in each partition as the "Base" block.

Ch ID	Characteristic	Base block	Block 1
C1	Capacity=>0	True: Capacity=>0	False: Capacity<0
C2	The input is a number	True: The input is a number	False: The input is not a number
C3	The input is not NaN	True: The input is not NaN	False: The input is NaN
C4	Pop the front element	True: Successfully pop an element from the queue	False: Fail to pop an element from the queue
C5	The queue is empty	True: The queue is empty	False: The queue is not empty
C6	The size of the queue is equal to the capacity	True: The queue is full	False: The queue is not full

(d). Define values for each block.

Ch ID	Characteristic	Base block	Block 1
C1	Capacity=>0	<b>True:</b> Capacity=>0	<b>False:</b> Capacity<0
C2	The input is a number	<b>True:</b> The input is a number	<b>False:</b> The input is not a number
C3	The input is not NaN	<b>True:</b> The input is not NaN	<b>False:</b> The input is NaN
C4	Pop the front element	<b>True:</b> Successfully pop an element from the queue	<b>False:</b> Fail to pop an element from the queue
C5	The queue is empty	<b>True:</b> The queue is empty	<b>False:</b> The queue is not empty
C6	The size of the queue is equal to the capacity	<b>True:</b> The queue is full	<b>False:</b> The queue is not full

(e). Define a test set that satisfies Base Choice Coverage (BCC). Write your tests with the values from the previous step. Be sure to include the test oracles.

Ch ID	Characteristic	BoundedQueue	enqueue	dequeue	is_empty	is_full	Partition (Base)
C1	Capacity=>0	O	O	O	O	O	{T, F}
C2	The input is a number		O				{T, F}
C3	The input is not NaN		O				{T, F}
C4	Pop the front element			O			{T, F}
C5	The queue is empty			O	O		{T, F}
C6	The queue is full		O			O	{T, F}

Methods	Ch ID	Base	Test Requirements	Infeasible TRs	#TRs
BoundedQueue	C1	{T}	{T, F}		2
enqueue	C1 C2 C3 C6	{TTTT}	{TTTF, TFTE, TTFF, TTTT}	{FTTF}	4
dequeue	C1 C4 C5	{TTF}	{TTF, TTT}	{FTF, TFF}	2
is_empty	C1 C5	{TT}	{TT, TF}	{FT}	2
is_full	C1 C6	{TT}	{TT, TF}	{FT}	2

After testing, we can find that the coverage is 100%.

file	line %	branch %	funcs %	uncovered lines
BoundedQueue_test.js	100.00	100.00	100.00	
BoundedQueue.js	100.00	100.00	100.00	
all files	100.00	100.00	100.00	